IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

IEEE Xplore®
1 Million Documents
1 Million Users

» Search Results

Help FAQ Terms IEEE Peer Review Quick Links

Welcome to IEEE Xplore®

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced
- CrossRef

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

IEEE Enterprise

- Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **51417** of **1131693** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

(network <and> (event <or> performance)) Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

16 **Performance-driven design and redesign of high-speed local area networks**
Ravikumar, C.P.; Pandit, D.R.; Mishra, A.;
High Performance Computing, 1998. HIPC '98. 5th International Conference On , 17-20 Dec. 1998
Pages:416 - 421

[Abstract] [PDF Full-Text (112 KB)] **IEEE CNF**

17 **New concepts of ATM network performance specifications**
Takahashi, K.; Yokoi, T.; Yamamoto, Y.;
TENCON 90. 1990 IEEE Region 10 Conference on Computer and Communication Systems , 24-27 Sept. 1990
Pages:532 - 536 vol.2

[Abstract] [PDF Full-Text (504 KB)] **IEEE CNF**

18 **Service disciplines performance for WWW traffic in GPRS system**
Ajib, W.; Godlewski, P.;
3G Mobile Communication Technologies, 2000. First International Conference on (IEE Conf. Publ. No. 471) , 27-29 March 2000
Pages:431 - 435

[Abstract] [PDF Full-Text (416 KB)] **IEE CNF**

19 **Evaluating the ability of SDH to transport broadcast quality video**
Lum, M.J.;
Broadcasting Convention, International (Conf. Publ. No. 428) , 12-16 Sept. 1996
Pages:576 - 582

[Abstract] [PDF Full-Text (572 KB)] **IEE CNF**

20 **Performance analysis of STC104 interconnection networks**

Hyo Jong Lee; Byeong Yeol Song;
High Performance Computing on the Information Superhighway, 1997. HPC Asia '97 , 28 April-2 May 1997
Pages:56 - 60

[Abstract] [PDF Full-Text (448 KB)] **IEEE CNF**

---

21 **Toward a more realistic performance evaluation of interconnection networks**
Ligon, W.B., III; Ramachandran, U.;
Parallel and Distributed Systems, IEEE Transactions on , Volume: 8 , Issue: 7 , July 1997
Pages:681 - 694

[Abstract] [PDF Full-Text (364 KB)] **IEEE JNL**

---

22 **A high performance communication subsystem for PODOS**
Vazhkudai, S.; Maginnis, T.;
Cluster Computing, 1999. Proceedings. 1st IEEE Computer Society International Workshop on , 2-3 Dec. 1999
Pages:81 - 91

[Abstract] [PDF Full-Text (276 KB)] **IEEE CNF**

---

23 **Fuzzy performance management of IEEE 802.4 token bus networks**
Sang-Ho Lee; Joon-Woo Son; Suk Lee;
American Control Conference, 1995. Proceedings of the , Volume: 5 , 21-23 June 1995
Pages:3254 - 3258 vol.5

[Abstract] [PDF Full-Text (496 KB)] **IEEE CNF**

---

24 **NOM-a tool for optimal design and performance evaluation of routing strategies and its application to the Telenet network**
Gersht, A.; Shulman, A.; Nemirovsky, P.;
INFOCOM '88. Networks: Evolution or Revolution? Proceedings. Seventh Annual Joint Conference of the IEEE Computer and Communcations Societies., IEEE , 27-31 March 1988
Pages:585 - 592

[Abstract] [PDF Full-Text (592 KB)] **IEEE CNF**

---

25 **Composite performance and availability analysis of wireless communication networks**
Yue Ma; Han, J.J.; Trivedi, K.S.;
Vehicular Technology, IEEE Transactions on , Volume: 50 , Issue: 5 , Sept. 2001
Pages:1216 - 1223

[Abstract] [PDF Full-Text (168 KB)] **IEEE JNL**

---

26 **Performance bounds for queueing networks and scheduling policies**
Kumar, S.; Kumar, P.R.;
Automatic Control, IEEE Transactions on , Volume: 39 , Issue: 8 , Aug. 1994
Pages:1600 - 1611

[Abstract] [PDF Full-Text (1060 KB)] **IEEE JNL**

---

27 **Failure dependent performance analysis of a fault-tolerant multistage interconnection network**
Kumar, V.P.; Reibman, A.L.;
Computers, IEEE Transactions on , Volume: 38 , Issue: 12 , Dec. 1989
Pages:1703 - 1713

[Abstract] [PDF Full-Text (920 KB)] **IEEE JNL**

---

28 **Network and service anomaly detection in multi-service transaction-based electronic commerce wide area networks**
Ho, L.; Papavassiliou, S.;
Computers and Communications, 2000. Proceedings. ISCC 2000. Fifth IEEE Symposium on , 3-6 July 2000
Pages:291 - 296

[Abstract] [PDF Full-Text (648 KB)] **IEEE CNF**

---

29 **Performance assessment of wavelength routed optical networks with shortest path routing over degree three topologies**
Coelho, R.M.F.; Rodrigues, J.J.P.C.; Freire, M.M.;
Networks, 2002. ICON 2002. 10th IEEE International Conference on , 27-30 Aug. 2002
Pages:3 - 8

[Abstract] [PDF Full-Text (466 KB)] **IEEE CNF**

---

30 **On worst case traffic in ATM networks**
Erimli, B.; Murphy, J.;
Twelfth UK Teletraffic Symposium. Performance Engineering in Telecommunications Networks (Digest No. 1995/054), IEE , 15-17 March 1995
Pages:15/1 - 1512

[Abstract] [PDF Full-Text (776 KB)] **IEE CNF**

Prev 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 Next

---

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 5 of 5 returned.

□ 1. Document ID: US 6671818 B1

Using default format because multiple data bases are involved.

```
   L6: Entry 1 of 5                         File: USPT                    Dec 30, 2003

US-PAT-NO: 6671818
DOCUMENT-IDENTIFIER: US 6671818 B1
** See image for Certificate of Correction **

TITLE: Problem isolation through translating and filtering events into a standard object format
in a network based supply chain

DATE-ISSUED: December 30, 2003

INVENTOR-INFORMATION:
NAME                             CITY            STATE     ZIP CODE       COUNTRY
Mikurak; Michael G.              Hamilton        NJ

US-CL-CURRENT: 714/4; 714/43, 714/48
```

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw Desc | Image

□ 2. Document ID: US 6606744 B1

```
   L6: Entry 2 of 5                         File: USPT                    Aug 12, 2003

US-PAT-NO: 6606744
DOCUMENT-IDENTIFIER: US 6606744 B1
** See image for Certificate of Correction **

TITLE: Providing collaborative installation management in a network-based supply chain
environment

DATE-ISSUED: August 12, 2003

INVENTOR-INFORMATION:
NAME                             CITY            STATE     ZIP CODE       COUNTRY
Mikurak; Michael G.              Hamilton        NJ

ASSIGNEE-INFORMATION:
NAME                 CITY            STATE     ZIP CODE      COUNTRY      TYPE CODE
Accenture, LLP       Palo Alto       CA                                   02

APPL-NO: 09/ 444654   [PALM]
DATE FILED: November 22, 1999

INT-CL: [07] G06 F 9/445
```

US-CL-ISSUED: 717/174; 717/174, 717/178, 705/26
US-CL-CURRENT: 717/174; 705/26, 717/178

FIELD-OF-SEARCH: 717/168, 717/170, 717/171, 717/174, 717/177, 717/172, 717/102, 717/176, 717/178, 705/1, 705/21, 705/26, 705/28, 709/201, 709/217, 709/227

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
4491947	January 1985	Frank	
4972453	November 1990	Daniel et al.	
5109337	April 1992	Ferriter et al.	
5159685	October 1992	Kung	
5297031	March 1994	Gutterman et al.	
5483637	January 1996	Winokur et al.	
5495610	February 1996	Shing et al.	709/221
5513343	April 1996	Sakano et al.	
5539877	July 1996	Winokur et al.	
5611048	March 1997	Jacobs et al.	713/202
5621663	April 1997	Skagerling	
5646864	July 1997	Whitney	
5655068	August 1997	Opoczynksi	
5694546	December 1997	Reisman	
5696975	December 1997	Moore et al.	717/168
5729735	March 1998	Meyering	
5761502	June 1998	Jacobs	
5764543	June 1998	Kennedy	
5768501	June 1998	Lewis	
5819028	October 1998	Manghirmalani et al.	
5832196	November 1998	Croslin et al.	
5864483	January 1999	Brichta	
5864662	January 1999	Brownmiller et al.	
5883955	March 1999	Ronning	
5890175	March 1999	Wong et al.	
5893905	April 1999	Main et al.	
5895454	April 1999	Harrington	
5907490	May 1999	Oliver	
5953707	September 1999	Huang et al.	
5974391	October 1999	Hongawa	
5974395	October 1999	Bellini et al.	705/9
5974403	October 1999	Takriti et al.	
5987423	November 1999	Arnold et al.	
5999525	December 1999	Krishnaswamy et al.	
6006016	December 1999	Faigon et al.	
6006196	December 1999	Feigin et al.	
6058426	May 2000	Godwin et al.	
6067525	May 2000	Johnson et al.	
6104868	August 2000	Peters et al.	

6105069	August 2000	Franklin et al.	709/229
6151582	November 2000	Huang et al.	
6157915	December 2000	Bhaskaran et al.	705/7
6167378	December 2000	Weber, Jr.	
6195697	February 2001	Bowman-Amuah	
6199204	March 2001	Donohue	717/178
6219700	April 2001	Chang et al.	709/222
6253339	June 2001	Tse et al.	
6256676	July 2001	Taylor et al.	709/246
6289462	September 2001	McNabb et al.	713/201
6314565	November 2001	Kenner et al.	717/171
6347398	February 2002	Parthasarthy et al.	717/178
6349237	February 2002	Koren et al.	
6470496	October 2002	Kato et al.	717/173
6487718	November 2002	Rodriguez et al.	717/177

OTHER PUBLICATIONS

Tan et al, "Applying component technology to improve global supply chain network management", ACM pp. 296-301, 1999.*
Ball et al, "Supply chian infrastructures system integration and information sharing", ACM SIGMOD, vol. 31, No. 1, pp. 61-66, Mar. 2002.*
Fu et al, "Multi agent enabled modeling and simulation towards collaborative inventory management in supply chains", ACM Proc. winter simulation, pp. 1763-1771, 2000.*
Zhao et al, "Data management issues for large scale distributed wokflow system on the internet", The database for Adv. in Inf. Sys. vo. 29, No. 4, pp. 22-32, 1998.*
"Network Trends: Internet Technology Improves Supply Chain Management". Asia computer Trends. Singapore. Dec. 14, 1998.
"Network Two Chooses Netcool to Support Ongoing Expansion and Proactive Management Initiative", Business Wire, Nov. 2, 1998, 2 pages, [Retrieved on Mar. 19, 2002], Retrieved from: Proquest.
"Proactive Networks Offers TelAlert-Pronto Watch 2.5 Integration", business Wire, Nov. 2, 1998, 2 pages, [Retrieved on Mar. 19, 2002], Retrieved from: Proquest.
"User's Guide for Microsoft Project." 1995; Microsoft Corporation. pp. 3,4,14-16, 82-84, 91, 130, 132-134, 175, 209. Document No. Pj62476-0895.

ART-UNIT: 2122

PRIMARY-EXAMINER: Khatri; Anil

ATTY-AGENT-FIRM: Oppenheimer Wolff & Donnelly, LLP Nader; Rambed

ABSTRACT:

A system, method and article of manufacture are provided for collaborative installation management in a network-based supply chain environment. According to an embodiment of the invention, telephone calls, data and other multimedia information are routed through a network system which includes transfer of information across the internet utilizing telephony routing information and internet protocol address information. The system includes integrated Internet Protocol (IP) telephony services allowing a user of a web application to communicate in an audio fashion in-band without having to pick up another telephone. Users can click a button and go to a call center through the network using IP telephony. The system invokes an IP telephony session simultaneously with the data session, and uses an active directory lookup whenever a user uses the system. Users include service providers and manufacturers utilizing the network-based supply chain environment.

18 Claims, 130 Drawing figures

□ 3. Document ID: US 6571285 B1

L6: Entry 3 of 5 File: USPT May 27, 2003

US-PAT-NO: 6571285
DOCUMENT-IDENTIFIER: US 6571285 B1

TITLE: Providing an integrated service assurance environment for a network

DATE-ISSUED: May 27, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Groath; Steve G.	Burnsville	MN		
Miller; Myke L.	Excelsior	MN		
Sachse; Christopher	Maplewood	MN		
Bloom; Jeremy D.	San Francisco	CA		
Turkson; Leslie T.	Eagan	MN		
Lund; Timothy	Lakeville	MN		
Beskar; Patrick J.	Mahotmedi	MN		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Accenture LLP	Palo Alto	CA			02

APPL-NO: 09/ 470776 [PALM]
DATE FILED: December 23, 1999

INT-CL: [07] G06 F 15/173

US-CL-ISSUED: 709/223; 709/223, 709/224, 370/352, 370/389
US-CL-CURRENT: 709/223; 370/352, 370/389, 709/224

FIELD-OF-SEARCH: 709/500, 709/527, 709/680, 709/683, 709/614, 709/615, 709/224, 709/318, 709/223, 370/352, 370/389, 370/392, 370/383, 370/395, 370/390

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5650994	July 1997	Daley	370/259
5774689	June 1998	Curtis et al.	709/500
5867495	February 1999	Elliott et al.	370/352
6182157	January 2002	Schlener et al.	

ART-UNIT: 2155

PRIMARY-EXAMINER: Winder; Patrice

ASSISTANT-EXAMINER: Nguyen; Thu Ha

ATTY-AGENT-FIRM: Oppenheimer Wolff & Donnelly LLP

ABSTRACT:

A method providing service assurance for a network to maintain an agreed upon Quality of Service. First, an alarm is generated to indicate a status of a network. The generation of the alarm comprises selecting a parameter of network to be monitored; determining a triggering level of the parameter; monitoring the parameter of an occurrence of the triggering level; and initiating alarm notification upon the monitored occurrence of the triggering level. Network event information is then dispatched upon generation of the alarm and is subsequently mapped. The data collected on the status of the network is then manipulated by concatenating the data collected on a network into a master file; reformatting the data into a standarized format; translating the data to key codes; sorting the data according to predetermined criteria; and concatenating the sorted data together. The data is then sorted in a database. Thereafter, network availability is conveyed graphically.

12 Claims, 39 Drawing figures

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. Desc | Image

□ 4. Document ID: US 6115393 A

L6: Entry 4 of 5 File: USPT Sep 5, 2000

US-PAT-NO: 6115393
DOCUMENT-IDENTIFIER: US 6115393 A

TITLE: Network monitoring

DATE-ISSUED: September 5, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Engel; Ferdinand	Northborough	MA		
Jones; Kendall S.	Newton Center	MA		
Robertson; Kary	Bedford	MA		
Thompson; David M.	Redmond	WA		
White; Gerard	Tyngsborough	MA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Concord Communications, Inc.	Marlboro	MA			02

APPL-NO: 08/ 505083 [PALM]
DATE FILED: July 21, 1995

PARENT-CASE:
CROSS REFERENCE TO RELATED APPLICATION This is a divisional of application Ser. No. 07/761,269 filed on Sep. 17, 1991, now abandoned, which is a continuation-in-part of U.S. patent application, Ser. No. 07/684,695, filed Apr. 12, 1991, now abandoned.

INT-CL: [07] H04 J 3/16, H04 J 3/22

US-CL-ISSUED: 370/469

US-CL-CURRENT: 370/469

•

FIELD-OF-SEARCH: 370/94.1, 370/85.13, 370/85.14, 370/94.2, 370/110.1, 370/79, 370/241, 370/252, 370/254, 370/465, 370/464, 370/466, 370/467, 370/469, 395/200, 395/183.15, 395/189.01, 395/189.04, 395/200.54, 395/285, 395/831, 371/20.1

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
4648061	March 1987	Foster	340/825.06
4817080	March 1989	Soha	
4887260	December 1989	Carden et al.	
4930159	May 1990	Kravitz et al.	
5021949	June 1991	Morten et al.	364/200
5025491	June 1991	Tsuchiya et al.	
5038345	August 1991	Roth	340/825.15
5060228	October 1991	Tsutsui et al.	370/85.13
5097469	March 1992	Douglas	
5101402	March 1992	Chiu et al.	
5136580	August 1992	Videlock et al.	370/85.13
5142528	August 1992	Kobayashi et al.	370/79
5142622	August 1992	Owens	395/200
5150464	September 1992	Sidhu et al.	395/200
5347524	September 1994	I'Anson et al.	371/20.1

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
WO 88/06822	September 1988	WO	

OTHER PUBLICATIONS

Hewlett-Packard brochure regarding local area network protocol analyzer (HP 4972A), Jun. 1987.
F. Kaplan et al., "Application of Expert Systems to Transmission Maintenance", IEEE, 1986, pp. 449-453.
D.M. Chiu et al., "Studying the User and Application Behaviour of a Large Network", Jun. 30, 1988, pp. 1-23.
R. Sudama et al., "The Design of a Realtime DECnet Performance Monitor", Jul. 15, 1988, pp. 1-23.
B.L. Hitson, "Knowledge-Based Monitoring and Control: An Approach to Understanding the Behavior of TCP/IP Network Protocols", ACM, 1988, pp.
210-221.
A.T. Dahbura et al., "Formal Methods for Generating Protocol Conformance Test Sequences", Proceedings of the IEEE, vol. 78, No. 8, Aug. 1990, pp. 1317-1326.
Hewlett-Packard Datasheet brochure, "Analyzing TCP/IP Networks with the HP 4972A", Sep. 1989, pp. 1-8.

ART-UNIT: 278

PRIMARY-EXAMINER: Patel; Ajit

ATTY-AGENT-FIRM: Fish & Richardson P.C.

ABSTRACT:

•

Monitoring is done of communications which occur in a network of nodes, each communication being effected by a transmission of one or more packets among two or more communicating nodes, each communication complying with a predefined communication protocol selected from among protocols available in the network. The contents of packets are detected passively and in real time, communication information associated with multiple protocols is derived from the packet contents.

25 Claims, 48 Drawing figures

| Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw Desc | Image |

□ 5. Document ID: US 5812533 A

L6: Entry 5 of 5 File: USPT Sep 22, 1998

US-PAT-NO: 5812533
DOCUMENT-IDENTIFIER: US 5812533 A

TITLE: Service provision in communications networks

DATE-ISSUED: September 22, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Cox; Richard D.	Garland	TX		
Hunter; Andrew T.	Dallas	TX		
Rand; Jeffrey K.	Coppell	TX		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
British Telecommunications public limited company	London			GB2	03

APPL-NO: 08/ 619660 [PALM]
DATE FILED: December 26, 1996

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
EP	94301397	February 28, 1994

PCT-DATA:

APPL-NO	DATE-FILED	PUB-NO	PUB-DATE	371-DATE	102(E)-DATE
PCT/GB95/00421	February 28, 1995	WO95/23483	Aug 31, 1995	Dec 26, 1996	Dec 26, 1996

INT-CL: [06] H04 L 12/16, H04 Q 11/00

US-CL-ISSUED: 370/259; 370/409, 455/4.2, 348/7
US-CL-CURRENT: 370/259; 370/409

FIELD-OF-SEARCH: 370/259, 370/399, 370/409, 370/432, 455/3.1, 455/4.2, 348/6, 348/7, 395/200.33, 395/200.49, 395/200.57, 395/235, 395/625, 395/680, 395/683

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5475819	December 1995	Miller et al.	395/200.33
5519443	May 1996	Salomon et al.	348/467
5621734	April 1997	Mann et al.	395/200.57

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
WO A 92 11724	July 1992	WO	

OTHER PUBLICATIONS

Wyatt et al, "British Telecom's Approach to the Intelligent Network", Second IEE National Conference On Telecommunications, York GB, p. 397.
Sivey, "Planning of Intelligent Network Services", Annual Review Of Communications, vol. 47, 1993-1994, Chicago, US. pp. 428-434.
Brosemer et al, "Virtual Networks: Past, Present and Future", IEEE Communications Magazine, vol. 30, No. 3, Mar. 1992, New York, US, pp. 80-85.
Desbiens et al, "Modeling and Formal Specification of the Personal Communication Service", IEEE INFOCOM'93, vol. 2, 28 Mar. 1992, San Francisco, US, pp. 756-765.
Malek et al, "On-Line Provisioning of Network Services", IEEE Journal of Selected Areas In Communication, vol. 6, New York, US, p. 662.
Stowe, "Service Management for the Advanced Intelligent Network", Globecom 91, Session 46, Paper 4, vol. 3, Phoenix, US, p. 1667.
Maeda et al, "An Intelligent Customer-Controlled Switching System", IEEE Global Telecommunications Conference & Exhibition 1988, Sessioin 46, Paper 1, vol. 3, Hollywood, FA US, p. 1499.

ART-UNIT: 272

PRIMARY-EXAMINER: Patel; Ajit

ATTY-AGENT-FIRM: Nixon & Vanderhye P.C.

ABSTRACT:

A communications network offers a variety of services to the customer while being able to add or modify the portfolio of services available. A service delivery infrastructure is provided, which would sit in the Service Control Point of an intelligent network architecture, and which delivers services using an array of service independent features. In the arrangement described, the service delivery infrastructure has an object oriented architecture and interacts with systems, such as billing and network management in the communications network by means of objects within the infrastructure. An aspect of the infrastructure is the provision of selected sets of services to users of the communications network, which selected sets effectively provide dedicated service networks to each customer.

14 Claims, 48 Drawing figures

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Image

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
BACKUP	90241
BACKUPS	3452
(5 AND BACKUP).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	5
(L5 AND BACKUP).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	5

Display Format: - Change Format

Previous Page Next Page Go to Doc#

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore® RELEASE 1.8

Welcome
United States Patent and Trademark Office

IEEE Xplore®
1 Million Documents
1 Million Users

» Search Results

Help FAQ Terms IEEE Peer Review

Quick Links

Welcome to IEEE Xplore®

○- Home
○- What Can I Access?
○- Log-out

Tables of Contents

○- Journals & Magazines
○- Conference Proceedings
○- Standards

Search

○- By Author
○- Basic
○- Advanced
○- CrossRef

Member Services

○- Join IEEE
○- Establish IEEE Web Account
○- Access the IEEE Member Digital Library

IEEE Enterprise

○- Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **6** of **1131693** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

(network <and> (event <or> performance)) <and> purge Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **A fault-tolerant multiprocessor cache memory**
Xiao Luo; Muzio, J.C.;
Memory Technology, Design and Testing, 1994., Records of the IEEE International Workshop on , 8-9 Aug. 1994
Pages:52 - 57

[Abstract] [PDF Full-Text (460 KB)] **IEEE CNF**

2 **Web caching in broadcast mobile wireless environments**
Katsaros, D.; Manolopoulos, Y.;
Internet Computing, IEEE , Volume: 8 , Issue: 3 , May-Jun 2004
Pages:37 - 44

[Abstract] [PDF Full-Text (439 KB)] **IEEE JNL**

3 **A Broadband Packet Switch for Integrated Transport**
Hui, J.; Arthurs, E.;
Selected Areas in Communications, IEEE Journal on , Volume: 5 , Issue: 8 , Oct 1987
Pages:1264 - 1273

[Abstract] [PDF Full-Text (968 KB)] **IEEE JNL**

4 **RMCM: reliable multicasts for core-based multicast trees**
Yuan Gao; Ge, Y.; Hou, J.C.;
Network Protocols, 2000. Proceedings. 2000 International Conference on , 14-17 Nov. 2000
Pages:83 - 94

[Abstract] [PDF Full-Text (1156 KB)] **IEEE CNF**

5 **Improved caching strategies in on-demand routing protocols for mobile ad hoc networks**

Wu Dong-ya; Hou Zi-feng; Hou Chao-zhen;
Communication Technology Proceedings, 2003. ICCT 2003. International Conference on , Volume: 2 , 9-11 April 2003
Pages:1258 - 1261 vol.2

[Abstract] [PDF Full-Text (545 KB)] **IEEE CNF**

6 **A ring purger for the FDDI token ring**
Yang, H.; Ramakrishnan, K.K.;
Local Computer Networks, 1991. Proceedings., 16th Conference on , 14-17 Oct. 1991
Pages:503 - 514

[Abstract] [PDF Full-Text (1124 KB)] **IEEE CNF**

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

□ Generate Collection Print

L7: Entry 4 of 5 File: USPT May 27, 2003

DOCUMENT-IDENTIFIER: US 6571285 B1
TITLE: Providing an integrated service assurance environment for a network

Abstract Text (1):
A method providing service assurance for a network to maintain an agreed upon Quality of Service. First, an alarm is generated to indicate a status of a network. The generation of the alarm comprises selecting a parameter of network to be monitored; determining a triggering level of the parameter; monitoring the parameter of an occurrence of the triggering level; and initiating alarm notification upon the monitored occurrence of the triggering level. Network event information is then dispatched upon generation of the alarm and is subsequently mapped. The data collected on the status of the network is then manipulated by concatenating the data collected on a network into a master file; reformatting the data into a standarized format; translating the data to key codes; sorting the data according to predetermined criteria; and concatenating the sorted data together. The data is then sorted in a database. Thereafter, network availability is conveyed graphically.

Brief Summary Text (8):
Despite the foregoing efforts, network failures are inevitable, and there is a need of monitoring network performance for the purpose of maintaining a predetermined agreed upon QoS.

Brief Summary Text (10):
A method providing service assurance for a network to maintain an agreed upon Quality of Service. First, an alarm is generated to indicate a status of a network. The generation of the alarm comprises selecting a parameter of network to be monitored; determining a triggering level of the parameter; monitoring the parameter of an occurence of the triggering level; and initiating alarm notification upon the monitored occurrence of the triggered level. Network event information is then dispatched upon generation of the alarm and is subsequently mapped. The data collected on the status of the network is then manipulated by concatenating the data collected on a network into a master file; reformatting the data into a standarized format; translating the data to key codes; sorting the data according to predetermined criteria; and concatenating the sorted data together. The data is then stored in a database. Thereafter, network availability is conveyed graphically.

Drawing Description Text (5):
FIG. 3 illustrates one embodiment of the present invention for dispatching network event information of a network with service assurance capabilities;

Drawing Description Text (34):
FIG. 31 illustrates an embodiment of the present invention which maps events on a network with service assurance capabilities; and

Detailed Description Text (17):
The development of graphical user interfaces began to turn this procedural programming arrangement inside out. These interfaces allow the user, rather than program logic, to drive the program and decide when certain actions should be performed. Today, most personal computer software accomplishes this by means of an event loop which monitors the mouse, keyboard, and other sources of external events and calls the appropriate parts of the programmer's code according to actions that the user performs. The programmer no longer determines the order in which events occur. Instead, a program is divided into separate pieces that are called at unpredictable times and in an unpredictable order. By relinquishing control in this way to users, the developer creates a program that is much easier to use. Nevertheless, individual pieces of the program written by the developer still call libraries provided by the operating system to accomplish certain tasks, and the programmer must still determine the flow of control

within each piece after it's called by the event loop. Application code still "sits on top of" the system.

Detailed Description Text (30):
One embodiment of the present invention is composed of multiple software programs which are linked together to create an architecture which is capable of monitoring a network for events and checking system functions and resources. Such events can include alarms, faults, alerts, etc. Other embodiments of the present invention may each include an individual software program.

Detailed Description Text (32):
Accordingly, FIG. 2 illustrates an embodiment of the present invention which provides service assurance for a network. In operation 200, an alarm is generated to indicate a status of a network. Network event information of the network is dispatched in operation 202 upon generation of the alarm after which the network event information is mapped in operation 204. The data collected on the status of the network is manipulated and stored in a database. See operations 206 and 208. In operation 210, availability of the network is conveyed graphically.

Detailed Description Text (38):
Accordingly, FIG. 3 illustrates one embodiment of the present invention for dispatching network event information of a network with service assurance capabilities. In operation 300, a network is monitored for an event. Thereafter, in operation 302, at least one notification action is generated based upon the occurrence of the event. The notification action may include an alphanumeric page, an e-mail message, a resolution script, a remedy trouble ticket, and/or a log message. Further, the notification action may be transmitted in operation 304 to notify a recipient about the occurrence of the event.

Detailed Description Text (45):
In another embodiment of the present invention, Data Acquisition scripts are programs which coordinate the collection and transfer of application logs to a central location. Data Acquisition can be used so that log files containing performance statistics gathered by a monitoring program can be transferred to a central server for processing by the Performance Data Manipulator (see below). The Data Acquisition scripts may be written in PERL

Detailed Description Text (47):
In an embodiment of the present invention, a PDM is a script that processes log files that have been collected by Data Acquisition in order to load the data into a database. The PDM converts the log files from formats specific to a particular monitoring program into a common format. PDM then formats the file based on data warehousing techniques which include converting nodes and performance metrics to key codes which are stored in the database. The coded data file is then bulk loaded into the database. The PDM may be written in PERL.

Detailed Description Text (62):
Automation scripts allow the automation of application and system management tasks. Automation scripts may monitor application health and perform corrective actions when the status of an application changes. Other functions may include sending SNMP traps when specific conditions are detected in an application or system.

Detailed Description Text (64):
One embodiment of the present invention provides the ability to correlate network events to individual customers (or providers in a Managed Network Services world) and notify customer service representatives of known outages affecting customers through a web interface. This allows proactive notification to customers of problems that affect them as well as builds confidence in customers calling to report problems that the provider is aware of.

Detailed Description Text (67):
Referring to FIG. 5, in one embodiment of the present invention, an activation signal is received in operation 500. Upon receipt of the activation signal, a signal is transmitted in operation 502 to initiate the retrieving of network performance data and network event data generated from at least one network monitor. Such network monitor is adapted for monitoring a network system and the relocating of the data into a common directory. Then, in operation 504, the signal is transmitted to initiate the manipulation of the data and the loading of the manipulated data into a database.

Detailed Description Text (70):
As an option, the present invention may further transmit a signal to initiate a clean archive program for deleting files from an archive directory, a signal to initiate a table extract program for extracting data from tables stored in the database, a signal to initiate a trigger reporting program for generating reports, a signal to initiate a purge record program for deleting records from the database, and a signal to initiate a database backup program for backing up data stored on the database.

Detailed Description Text (71):
The following subsections describe an embodiment of the present invention that controls the collection, manipulation and storage of network performance data and network event data of a network with service assurance capabilities and provides an exemplary step-by-step overview of the flow of data from collection to when it's loaded into the database. FIG. 6 is a flowchart that provides an overview of a data collection process of one embodiment of the present invention.

Detailed Description Text (74):
The data collection is started by the network monitory applications creating their ASCII text data files. These files are generally stored locally on the machines they are running on. Specifics on where these files should be stored are located in the installation & configuration instructions for each application.

Detailed Description Text (96):
The key_codes.pl script 624 uses perf_metric_tb.ext and network_element_tb.ext to look up the codes associated with the node or element being monitored. It assigns the key code as the name of the file (i.e. keycode.element.stage2 and keycode.element.stat). It also produces perf_metric_time_tb.dat.

Detailed Description Text (104):
6. purge_records.pl 634

Detailed Description Text (105):
7. backup.pl 636

Detailed Description Text (124):
Order of Scripts Called 1. Cleanup Utility 608 2. Data Acquisition Utility 610 3. Table Extract 616 4. Data Manipulation 618 5. Data Loader 632 6. TPSA_ProcessTheBatchQueue.txt (Reporting SPSS) 638 7. Purge Records 634 8. Backup 636

Detailed Description Text (145):
The key_codes.pl script uses perf_metric_tb.ext and network_element_tb.ext to look up the codes associated with the node or element being monitored. It assigns the key code to the name of the file in place of the element name (i.e. <keycode>.element.stage2 and <keycode>.element.stat). It also produces perf_metric_time_tb.dat, which is a list of all of the unique date-time stamps appearing in the raw data and their corresponding epoch times (the number of seconds since Jan 1, 1970)

Detailed Description Text (223):
Accordingly, FIG. 10 depicts an embodiment of the present invention which generates an alarm to indicate a status of a network for service assurance purposes. Such purposes can include identifying errors and faults, monitoring system resources, anticipating problems, etc. Once a parameter of a network that is to be monitored is selected in operation 1000, a triggering level of the parameter is determined in operation 1002. In operation 1004, the parameter for an occurrence of the triggering level is monitored. If the triggering level is reached, an alarm is initiated in operation 1006.

Detailed Description Text (224):
In one aspect of the present invention, the alarm is named for identification purposes. Optionally, the parameter can be adapted to be monitored for a plurality of components of the network. In such an embodiment, a separate alarm notification may be initiated for each of the components upon the monitored occurrence of the triggering level thereon.

Detailed Description Text (231):
Adding properties to an existing property group enables one to take advantage of the flexibility of property groups. Properties may need to be added if it is desired to: Make a property group unique by adding a property that isn't in any other group. By assigning the unique property group to selected devices, you can manage those devices differently from other devices. Make polls and/or alarms apply to devices belonging to that property group. Monitor a base object on a group of devices when the base object isn't listed as a property in the devices' property group.

Detailed Description Text (265):
Database Backup/Recovery

Detailed Description Text (266):
Backups of the SA database will meet the following requirements: The backup script utility may be written in perl. Perl allows scripts to be more portable across hardware platforms. The backup utility can be executed by the UNIX cron utility. It could also be executed by the Windows NT AT scheduler if running on the NT platform. Cold_backups will be run at least once a week. This requires database down time. Exports of the database will take place every evening at the end of the nightly batch schedule.

Detailed Description Text (279):
Phase 2 recommends detailed level data to be retained for 40 days and daily rollup data retained for 13 months. A perl script will be written to purge this data once is has expired. This process will run nightly, before the actual loading of the data occurs.

Detailed Description Text (337):
Database Cold Backup

Detailed Description Text (338):
This section will describe the required files needed to run cold_backups. It will outline the recommended steps to configure cold_backups (written in Perl) to run on a database. Finally, this section will discuss how to execute the script. The location of the cold_backups script can be anywhere in the database's directory system.

Detailed Description Text (340):
The following list of files and directories need to be created before cold_backups will run. These files are described bellow and common naming conventions are given. Also, an OFA directory structure location is specified.

Detailed Description Text (342):
This file should contain a list of one or more system ids, dbse home, and dbse base. Every system id will be shutdown, backed up, and restarted. The values of this file are also used to set the correct environment of the SID. The format of this file must be of the form: SID,DBSE_HOME,DBSE_BASE. There are no spaces between fields. Also, it is important that there are no blank lines in this file. A common naming convention for this file is cold_sids. The file is best housed with other files related to the backup.

Detailed Description Text (347):
Backup Directory

Detailed Description Text (348):
The backup directory will store the backup files. The naming convention for this directory is {SID}_cold where SID is the system id. The file is best housed in the current directory (described below in "Current Directory").

Detailed Description Text (350):
Directory paths and file names are hard coded in the cold_backups script. Therefore it is recommended to configure these variables to suit the file structure of the particular database the backup is to be performed on. The following are variables that need to be specified for the system in which the script is run on. These variables are grouped together at the beginning of the script. Look for the commented section entitled "Variable List" to locate the variables within the script.

Detailed Description Text (352):
This variable is set to the directory location in which the backup directory (described above in "Backup Directory"), log file (described below in "Log"), system id file (described above in System Identification (SID) File), and temporary directory (created during execution of the backup and deleted after its completion) are kept. This directory can be placed anywhere.

Detailed Description Text (358):
This file contains messages and errors generated from cold_backup. A common naming convention is cold.log. The log file is created in the current directory (see section 4.2 above) by cold_backups. The log file will be kept in an archived form, also in the current directory, for one previous backup and it will be named *.old where * represents the filename you designate as the log file.

Detailed Description Text (362):
This variable is set to a command line mail program. The mail program is used to notify the Database Administrator of problems occurred during execution of the backup or to inform the Administrator of a successful backup.

Detailed Description Text (369):
The cold_backups script can be run manually by having a user execute the script. Since the backup requires that the database be shutdown, inform all database users of the downtime. Expect up to fifteen minutes for the backup. In order to run the script manually go to the directory where the script is located. Check the permissions, and set if necessary. At the command line, type cold_backups. The script will run without user interaction. The script is complete when the prompt returns.

Detailed Description Text (371):
It is possible to schedule the backup using the UNIX crontab. To create a new crontab or to edit an existing one, type crontab -e at the UNIX command line. To list the entries in the crontab, type crontab -l at the command line. Entries in the crontab file are lines of six fields separated by spaces or tabs. An asterisk (*) in fields one through five indicates all legal values. Values can be separated by commas (for a list of values) or hyphens (for inclusive range). The fields are as follows:

Detailed Description Text (372):
By using the crontab, it is possible to schedule daily, weekly, or monthly backups during non-peak hours. Since the backup requires that the database be shutdown, inform all database users of the expected downtime. Expect up to fifteen minutes for the backup. Check the permissions of cold_backups before the first backup and set if necessary.

Detailed Description Text (374):
The following is an entry of a crontab. The cold_backups script will be backed up at. 2:30am daily. /files2/db/dev/backup1/cold_backups>;crontab-1 30 2* * * /files2/db/vendor/dbse/admin/general/unix/cold_backups/cold_backups

Detailed Description Text (375):
If both fields are specified in an entry, they are cumulative. For example: 0 0 1,15*1/files2/db/vendor/dbse/admin/general/unix/cold_backups/cold_backups

Detailed Description Text (376):
This crontab entry runs at midnight on the first and fifteenth of each month, as well as every Monday. To specify days in only one field, set the other field to asterisk (*). For example: 0 0 * *1/files2/db/vendor/dbse/admin/general/unix/cold_backups/cold_backups

Detailed Description Text (378):
Database Export Backup

Detailed Description Text (379):
This section will describe the files needed to run export_backups. It will outline the recommended steps to configure export_backups (written in Perl) to run on a database. Finally, this section will discuss how to execute the script. The location of the export_backups script should be in admin/general/unix (following OFA standards).

Detailed Description Text (380):
The export_backups script performs a full database export on a database by using an export utility. This script is meant to run daily and the backup files are stored in a cycle that can be configured to suit the needs of the client. For an explanation of the backup cycle and naming convention, see the sections below entitled "Configuration of Backup Cycle" and "How to Execute."

Detailed Description Text (382):
The following list of files and directories need to be created before export_backups will run. These files are described bellow and common naming conventions are given. Also an OFA directory structure suggestion is given.

Detailed Description Text (386):
This file will contain the current day number of the backup cycle. The backup cycle is commonly four weeks in length but it can be set at any interval (see section five below on configuration of cycle variables). The day file will contain an integer between one and the maximum day number of the cycle (29 for a four-week cycle). It is important that there are no blank lines in this file. The user will need to create this file before the initial execution of the backup and after that the file will update itself. It is a good idea to have the day number correspond to the current weekday. For example, a one in the day file will correspond with Sunday and a four will correspond with Wednesday. A common naming convention is day_file.dat. This file is best housed in the current directory (described below in "Current Directory").

Detailed Description Text (389):
Backup Directory

Detailed Description Text (390):
The backup directory will store the backup files. The naming convention for this directory is {SID}_exp where SID is the system id. The file is best housed in the current directory (described below in "Current Directory ").

Detailed Description Text (392):
Directory paths and file names are hard coded in the export_backups script. Therefore it is recommended to configure these variables to suit the file structure of the particular database the backup is to be performed on. The following are variables that need to be specified for the system in which the script is run on. These variables are grouped together at the beginning of the script. Look for the commented section entitled "Variable List" to locate the variables within the script.

Detailed Description Text (394):
This variable is set to the directory location in which the backup directory (described above in "Backup Directory"), log file (described below in "Log"), SID file (described above in "System Identification (SID) File"), and day file (described above in "Day File") are kept. This directory can be placed anywhere.

Detailed Description Text (400):
This file contains messages and errors generated from export_backup. A common naming convention is export.log. During execution, the log file is opened and the entries will be appended to the existing file. If the file does not exist, the script will create a new log file with the name you designate here. Old log files will be kept in an archived form, also in the current directory, and will contain the logged entries for an entire backup cycle. The archived file will be named *.yyyymmmdd (as in 2001Dec31) where * represents the filename you designate as the log file. The archive will occur if the day file contains the last day of the cycle.

Detailed Description Text (404):
This variable is set to a command line mail program. The mail program is used to notify the Database Administrator of problems occurred during execution of the backup or to inform the Administrator of a successful backup.

Detailed Description Text (407):
Configuration of Backup Cycle

Detailed Description Text (408):

The backup cycle can be changed to suit the particular requirements for the database. There are six cycle variables in the script that need to be configured if the cycle is changed. They are located in the top portion of the script in a section entitled "Cycle Variables" immediately following the "Variable List" described above in section four. The values of these variables are used to name the backup file and to determine the schedule for when the backup files rollover.

Detailed Description Text (410):
This variable should be set to the maximum number of daily backup files that are kept minus one. For example, if seven daily backups are kept, this value should be set to six. The naming scheme for the daily backup files is such that day00 is the current day's backup and day06 is six days ago.

Detailed Description Text (412):
This variable is a string and should be set to the three letter abbreviation of the week day name in which you want the weekly backup files to rollover. For example, if you want the week backup files to roll over on Sundays you would set this variable to "Sun".

Detailed Description Text (414):
This variable should be set to the maximum number of weekly backup files that are kept minus one. On the week ending date, the weekly backup files will be rolled back and day07 will become week01. For example, if you want to keep four weekly backup files, this variable should be set to three.

Detailed Description Text (416):
This variable should be set to the total number of days in the backup cycle plus one. The value of this variable will be the largest possible value that is in the day file (see section 3.2 above). For a four-week cycle, the maximum cycle days variable should be set to twenty-nine (4 weeks*7 days per week+1 day=29 cycle days). A value of fifty-seven would represent an eight week cycle (8*7+1=57). When the value in the day file is the same as the maximum cycle days variable the cycle backup will occur.

Detailed Description Text (418):
Weekly backups will be kept in longer cycle increments for long term storage. The value of this variable should be set to the increment value. This variable comes into effect when rolling back the week files (described in section 5.3). For example, if you want to keep backups in four-week intervals (starting at the last weekly backup) you would set this variable to four.

Detailed Description Text (420):
This variable should be set to the maximum number of backup files that are kept minus the cycle increment value. For example, if you want to keep backup files for fifty-two weeks (one year) and your increment value is four, the value of this variable should be forty-eight. If the day in the day file is equal to the maximum cycle days, the weekly backup files will be rolled back in increments designated by the increment cycle variable. For example, week04 would become week12 if the increment value were eight.

Detailed Description Text (423):
The export_backups script can be run manually by having a user execute the script. Since the backup requires that the database be shutdown, inform all database users of the downtime. Expect up to fifteen minutes for the backup. In order to run the script manually go to the directory where the script is located. Check the permissions, and set if necessary. At the command line, type export_backups. The script will run without user interaction. The script is complete when the prompt returns.

Detailed Description Text (425):
It is possible to schedule the backup using the UNIX crontab. To create a new crontab or to edit an existing one, type crontab -e at the UNIX command line. To list the entries in the crontab, type crontab -l at the command line. Entries in the crontab file are lines of six fields separated by spaces or tabs. An asterisk (*) in fields one through five indicates all legal values. Values can be separated by commas (for a list of values) or hyphens (for inclusive range). The fields are as follows:

Detailed Description Text (426):

By using the crontab, it is possible to schedule daily, weekly, or monthly backups during non-peak hours. Since the backup requires that the database be shutdown, inform all database users of the expected downtime. Expect up to fifteen minutes for the backup. Check the permissions of export_backups before the first backup and set if necessary.

Detailed Description Text (466):
Purge Records Script

Detailed Description Text (467):
The script purge_records is used to delete records from a Database. The script runs a PL/SQL script that deletes records from tables after a certain length of time. This script can be run from the cron or from the command line. The script performs error checking and reports via email to a specified administrator of success or failure. This script may be written in Perl.

Detailed Description Text (472):
A directory structure should be set up to house the files generated by purge_records. See section 0 for a list and description of these files. ##STR1##

Detailed Description Text (474):
The files used/created by the purge_records script should have a standardized naming format. The log file generated by the purge_records script will be named purge.log. This file will hold a months worth of log entries. At the first day of the month, it will be renamed purge.log.date_stamp. The current format of the $Date_stamp is MonYYYY, however $Date_stamp can be changed if more frequent archiving of the log file is necessary.

Detailed Description Text (475):
The file generated by the SQL Script is called purge.sql.log. This file is the spooled output of the SQL Script. It will be checked for errors/exceptions. If none are found, it is deleted. If there are errors found, it is renamed purge.sql.log.date_stamp with the date stamp format as described above.

Detailed Description Text (477):
This script uses a SQL script called purge_records. The scripts should be located in its own directory (following the OFA directory structure). Set $SQL_dir to this directory location. The SQL script used by purge_records is as follows:

Detailed Description Text (479):
Files Generated purge.sql.log (log file generated by SQL script purge_records) purge.log (log files generated by this script)

Detailed Description Text (481):
The purge_records script will send email to an administrator to report success or failure of the purging process. Set $Mailer to the text string that will execute the mailer in command line mode. Set $DBA to the email address of the administrator.

Detailed Description Text (483):
The purge_records script will need to log on to SQL*Plus to execute the SQL script purge_records. A variable name should be set to the schema that owns the tables that are going to be purged. Do not hard-code the schema password in the script. Instead use getpass.

Detailed Description Text (484):
Executing purge_records

Detailed Description Text (485):
The purge_records script does not accept any parameters. It can be run from the command line or from the cron. To execute the script, type purge_records.

Detailed Description Text (487):
The following is the procedure the purge_records script follows to load data into a database.

Detailed Description Text (489):
The following is the scripts reference above are used by purge_records. The bolded lines should be configured to the particular system you are using.

Detailed Description Text (506):
The following is the script reference above this is used by purge records. The bolded lines should be configured to the particular system you are using.

Detailed Description Text (509):
FIG. 16 depicts an embodiment of the present invention which graphically conveys availability in a network with service assurance capabilities. In operation 1600, report parameters are selected relating to availability of monitored elements, services, and processes of a network. A database is polled in operation 1602 for data that matches the report parameters. A graph is generated in operation 1604 from the data that matches the report parameters. In operation 1606, the generated graph is displayed to graphically represent the monitored elements, services, and processes of the network.

Detailed Description Text (528):
To enable report generation, execute IPSA_StartSPSS.cmd. Both SPSS and a VB file-polling application are invoked in the form of IPSA_SPSS.exe, sppssw.exe, and spsswin.exe. These are processes that can be monitored using Window NT Task Manager. With report criteria specified in <SABATCH>IPSA_BatchQueue.txt, create the file <SAADHOC>IPSA_ProcessTheBatchQueue.txt. The contents of this file are unimportant, as only its presence is polled for. IPSA_SPSS.exe reads the contents of IPSA_BatchQueue.txt and passes it to SPSS. Graphs are generated in <SAWEB>.backslash.Batch.backslash.Daily.backslash. as well as an HTML file that points to the images.

Detailed Description Text (553):
FIGS. 19-22 provide historical record of collected performance data and network events. Exception reporting is limited to views of events that occurred in real-time and does not include finding exceptions in the historical data by analyzing past data.

Detailed Description Text (844):
At--Examples To display a list of commands scheduled on the server MARKETING, type at .backslash..backslash.marketing To learn more about a command with the identification number 3 on the server CORP, type at .backslash..backslash.corp 3 To schedule a net share command to run on the CORP server at 8:00 A.M., and redirect the listing to the server MAINTENANCE, shared directory REPORTS, and file CORP.TXT, type at .backslash..backslash.corp 08:00 "cmd /c net share
reports=d:.backslash.marketing.backslash.reports>> .backslash..backslash.maintenance.backslash.r
To back up the MARKETING server's hard disk to a tape drive at midnight every five days, create a batch program (ARCHIVE.CMD) containing the backup commands. Then schedule the batch program to run by typing at .backslash..backslash.marketing 00:00 .backslash.every:5,10,15,20,25,30 archive To cancel all commands scheduled on the current server, clear the at schedule information by typing at .backslash.delete

Detailed Description Text (888):
Lastly is the `Poll Condition` tab. This is straightforward to use, and consists of choosing the necessary base object and checking the attributes that you would like to monitor. To do the very minimum, and grab variables for entry into logs, you would simply select the attribute in question and select `present` (if more select `AND` then select the next attribute and `present` and so on).

Detailed Description Text (919):
For application and server monitoring, the patroller is used. This creates a monitoring layer that resides on individual systems in the form of agents. These agents poll for user-defined information on pre-selected intervals and typically push this data up to a central point of control. From this point of control the data can be collated and correlated to provide historical data on the performance of the systems in the environment.

Detailed Description Text (961):
FIG. 31 illustrates an embodiment of the present invention which maps events on a network with service assurance capabilities. In operation 3100, a network is monitored for the occurrence of availability events, threshold events, and trap events. At least one occurred event is correlated to at least one other occurred event in operation 3102 to generate at least one correlating event. In operation 3104, the occurred events and correlating events are mapped on

at least one network map. The network map is subsequently displayed in operation 3106.

Detailed Description Text (962):
In one embodiment of the present invention, the step of monitoring the network further comprises: tracking the availability of individual components of network for events, tracking the availability of individual services of the network for events, tracking the availability of individual processes of an operating system of the network for events, tracking the status of agent processes on individual components of the network for events, monitoring the operating system and application performance of network for threshold events, and monitoring traps of the network for events.

Detailed Description Text (963):
In yet another embodiment, the network map is a node level map and/or an event level map. The node level map displays node responding events, agent not responding events, and/or node down events. The step of mapping the occurred events and correlating events when the network map comprises the event level map further comprise of additional steps. In particular, the occurred events and correlating events may be filtered based upon predetermined criteria. The filtered events may also be mapped on the event level map. In still yet another embodiment, at least one notification action is generated based upon the occurred events and/or correlating events.

Detailed Description Text (965):
This section details the three availability events monitored. Node Up and Node Down/Interface Up and Interface Down--Tracking an individual network component (such as a router, server, workstation, etc.). This will be tracked using Network Node Manager and ECM. We will track all nodes which we monitor on the test network, including the following: nsmmws16, nsmmws09, twmmnt02, twmmdb02, nsmmrt03, nsmmrt04. When a Node or Interface fails to respond to a ping, a Node or Interface Down event for the specific node will be generated. When a Node or Interface responds to a ping after immediately after failing to respond to a ping, a Node or Interface Up for the specific node event will be generated.

Detailed Description Text (966):
Service Up and Service Down--Tracking an individual network service (FTP, NNTP, POP3, SMTP, DNS, HTTP, and RADIUS (RADIUS is tracked as a probe only)). This will be tracked using the Collector Internet Service Monitor. When a service fails to respond to the ISM a Service Down event for the specific service will be generated. When a Service responds to the ISM immediately after failing to respond to the ISM, a Service Up event for the specific service will be generated.

Detailed Description Text (970):
This section details the operating system threshold events and the application performance threshold events monitored. Patroller will be used to monitor processes and send a Parameter High event for a specific parameter and node when a parameter crosses a threshold by increasing its value and send a Parameter Low event for a specific parameter and node when a parameter crosses a threshold by decreasing its value.

Detailed Description Text (971):
The following parameters should be monitored for Phase 2. Note that these parameters are for proof of concept only, as parameters can be added or removed very easily.

Detailed Description Text (973):
This section details the operating system threshold events and the application performance threshold events monitored. ECM will be used to monitor SNMP variables and send a Parameter High event for a specific parameter and node when a parameter crosses a threshold by increasing its value and send a Parameter Low event for a specific parameter and node when a parameter crosses a threshold by decreasing its value.

Detailed Description Text (974):
The following parameters should be monitored and possibly more:

Detailed Description Text (976):
This section details the requirements for monitoring any MIB defined trap from any monitored SNMP device. The SNMP trap will by generated asynchronously by any monitored node and will be translated into an event.

Detailed Description Text (984):
This section details the requirements for the network maps. There will be two network maps: Node Level Map--a node level map may be provided by HP Open View Network Node Manager. It will display each managed in one of three different colors, corresponding to Node Responding, SNMP Agent Not Responding, and Node Down. Event Level Map--an event level map can be provided by Collector Objective View. It will display entities driven by arbitrary event filters, but will not contain a node level view. These entities will be set to monitor specific groups of events and will change color to match the most critical severity of any event in the event filter.

Detailed Description Text (992):
Collector Process Control is used to start and stop Collector components and monitor their execution. It allows for a single point of control for the components of the system. The process control system contains the following elements: Process Control Agents, which are programs installed on each host with the responsibility of managing processes. A set of command line utilities to provide an interface to process management

Detailed Description Text (1015):
Collector Internet Service Monitoring (ISM)

Detailed Description Text (1016):
This section will discuss the Service Assurance test environment specific details of the Collector Internet Service Monitors (ISM's) and requirements for a remote installation.

Detailed Description Text (1017):
Internet Service Monitors Remote Installation

Detailed Description Text (1024):
The Collector/etc directory is where the interfaces file (discussed bellow) will be placed. The Collector/var directory is where all event data is kept while any Monitors are in Store and Forward mode (when the ObjectServer is off-line).

Detailed Description Text (1025):
Process Control will insure that the Monitors are restarted in the event they unexpectedly go down.

Detailed Description Text (1029):
Internet Service Monitoring Configuration

Detailed Description Text (1031):
The on-line configuration tool, if run from a web server, requires the start java.sh script (/opt/Collector/monitors/config) to be running. This script has a timeout line, which by default is set to ten minutes (in seconds). If the on-line configuration tool is idle for more than ten minutes, a server error will be generated the next time the tool is requested. Running the script from the command line is required before using the tool again. For a work around, the timeout value has been set to 3600 seconds (one hour) to increase the time the configuration tool can be accessed.

Detailed Description Text (1033):
Store and forward is a function that allows monitors and probes to record all event messages to an ObjectServer.store file while the Objectserver is down. This file will be located in the Collector/var directory.

Detailed Description Text (1034):
The store and forward function is set to 0 (off) by default with the ISM's. The store and forward function has been turned on for the ISM's installed on nsmmws09 by using VI to set store and forward to 1 (on). This is done in the monitor.props files (/opt/Collector/monitors/solaris2).

Detailed Description Text (1036):
The Web Servers on twmmnt02 (Microsoft IIS) and nsmmws09 (Netscape Fast-Track Server) are being monitored. The polling interval is set at fifteen minutes.

Detailed Description Text (1110):
Requirements for the Development Environment The ability to check in and check out files so that only one person is editing a file at a time The following will be tracked on each file that is being version controlled. date of creation, date of last modification, version, and change history (annotated with date, rev, user, and comments). The ability to associate revision numbers with development environment (the environment can be set to `dev`, `tst`, or `prd`). This make is it make it possible to develop multiple releases at one time. The ability to retrieve previous versions of a component or sub-component for either edit or review. Support for multiple development languages. Ability for users to operate in a separate environment. This includes operations on the users `own` test data and executables. Backup and recovery of source code, documentation, test data, etc... Tools to aid in the debugging of components and sub-components. This would include generation of test data and unit test conditions. Documentation for users on how to use the tools under different conditions and situations. Ability to tie SIR or defect number to all components and sub-components that are. Ability to migrate components and sub-components between environments. Documented coding standards for each type of development language used). Provide shells as a starting point for each coding language used. Strategy for software distribution.

Detailed Description Text (1129):
Determine Backup/Recovery Requirements

Detailed Description Text (1130):
A person should be designated to receive nightly backup confirmations. Two confirmations are sent per machine, one is for full backup and the other is for incremental backup. Incremental backups may be run every night and full backups can be run once a week between Friday evening and Monday morning.

Detailed Description Text (1139):
Determine Backup/Recovery Requirements

Detailed Description Text (1140):
This section will list Service Assurance's responsibilities to an exemplary network to insure timely backup and recovery.

Detailed Description Text (1142):
Changes to monitoring and backups should be logged as a trouble ticket with the computer related service center.

Detailed Description Text (1143):
An individual from the project should be identified to receive backup completion notices. These notices are mailed at the completion of each nightly backup cycle. This person should then verify that all Service Assurance servers were adequately backed up the previous night.

Detailed Description Text (1162):
External disk storage will vary based on the size of the environment being monitored. The 20 GB listed above should be sufficient for a medium sized environment (.about.500 monitored nodes).

Detailed Description Paragraph Table (30):
TABLE 13 Script Name Type Input Output Functionality purge records script none purge.sql.log Deletes rows ds of tables that are past the specified date (section 0)

Detailed Description Paragraph Table (31):
TABLE 14 Step Activity/Action 1 Set database environment Set directory structure variables Set SQL Script directory Set log file name variables 2 Create a new log file purge.log if one does not exist, or open an existing purge.log for appending 3 Set the mailer program and administrator's email address 4 Set the date stamp for load For the schema owner, set the user name and get password using getpass Set the fail flag to false (0) 5 Log onto SQL*Plus Execute the SQL script purge records 6 Check to see if a the SQL script generated a log file If so, scan the log file for errors. If errors are found set the fail flag to true. Print to the log file the number of rows that were deleted If the log file does not exist, the script did not run correctly. Set the fail flag to true. 7 Check the value of the failure flag (it will be set to true if any errors occurred at any time during the process above) If the flag is false the purge was successful. Delete the log file generated by the SQL script and send an email report

to the administrator If the flag is true the purge was unsuccessful. Rename the file generated by the SQL script to include the date stamp and send an email report to the administrator 8 If it is the first day of the month, rename purge.log to include the date stamp. Otherwise print the date/time to the log and close it

Detailed Description Paragraph Table (32):
getpass (Born Shell) #!/bin/sh #----------------------# #Scipt_name: get_pass # # # #Description: This script wiIl be used to retrieve the # #appropriate password from the password # #file. It can be used from the command # #line to retrieve a password or from a # #shell script to eliminate hard coding of # #passwords. The .password file is located # #in $DBSE_HOME, with the executable # #located in $DBSE_HOME/bin # #Dependencies: get pass requires one file, password # #SID_FILE: Contains a list of passwords for # #system, sys, and dbse. # #Command syntax: getpass USERNAME # #-------------------# #what shell do we use 48 SHELL = /usr/bin/sh #where is our home? DBSE_HOME =/files0/ipsa/vendor/dbse/product/ 7.3.4; export DBSE_HOME #what path do we look for? PATH=/usr/bin:/bin:${DBSE_HOME}/bin; export PATH #who are we retrieving the password for if[$1] then USER =$1 else echo "" echo "USAGE: getpass 'username'" echo"" exit 1 fi #make sure the password file exist if[!-s $DBSE_HOME/.password] then echo echo "ERROR: this machine does not appear to have a password file" echo "" exit 1 fi #get the appropriate password PASSWORD='cat $DBSE_HOME/.password.vertline.grep-i "${USER}".vertline. awk'{print $2}" BAD_SIDS ='cat $DBSE_HOME/.password.vertline.grep-i "${USER}".vertline. awd '{print $3}" OLD_PASS='cat $DBSE_HOME/.password.vertline.grep -i"${USER} ".vertline. awk '{print $4}" #if a password was found, print it to the screen if [$PASSWORD] then echo "${PASSWORD}" #print a message if passwords appear to not be synced between databases if[$BAD_SIDS] then BAD_SIDS ='echo ${BAD_SIDS}.vertline.cut -c2-100' echo "" echo "WARNING:\tThe password for ${USER} may not be synchronized" echo "\t\tbetween all databases. The database(s) ${BAD_SIDS} appear(s)" echo "t\tto use the old password '${OLD_PASS}'" echo "" fi else echo " " echo "ERROR: $ {USER} is not a supported username" echo " " exit 1 fi purge_records (SQL Script) SET ECHO ON SBT FEEDBACK ON SET FLUSH OFF SET HEADING OFF SET SERVEROUTPUT ON SET TERMOUT OFF SET VBRIFY OFF SPOOL /fiIes6/ipsa/purge/purge.sql.log REM *purge records in PERF_FACT_TB table delete from PERF_FACT_TB where PERF_FACT_TB.PERF_TIME_KEY_CD in (select PERF_METRIC_TIME_TB.PERF_TIME_KEY_CD from PERF_METRIC_TIME_TB where to_char(PERF_METRIC_TIME_TB.PERF_DT, 'MM/DD/YYYY') <=(select to_char (sysdate - 40, 'MM/DD/YYYY') from dual)); REM *Purge records in EVENTS_FACT_TB table delete from EVENTS_FACT_TB where EVENTS_FACT_TB.PERF_TIME_KEY CD in (select PERF_METRIC_TIME_TB.PERF_TIME_KEY_CD from PERF_METRIC_TIME_TB where to_char (PERF_METRIC_TIME_TB.PERF_DT, 'MM/DD/YYYY') <=(select to_char(sysdate -40, 'MM/DD/YYYY') from dual)); RBM *purge records in PERF_FACT_DLY_TB table delete from PERF_FACT_DLY_TB where PERF_FACT_DLY_TB.PERF_TIME_KEY_CD in (select PERF_METRIC_TIME_TB.PERF_TIME_KEY CD from PERF_METRIC_TIME_TB where to_char(PERF_METRIC_TIME_TB.PERF_DT, 'MM/DD/YYYY') <=(select to_char (sysdate - 397, 'MM/DD/YYYY') from dual)); SPOOL OFF

Detailed Description Paragraph Table (56):
STATUS ID Day Time Command Line 0 Each F 04:39 PM net send group leads status due 2 Each M 12:00 AM chkstor>check.fil 3 Each F 11:59 PM backup2.bat

Detailed Description Paragraph Table (58):
at 1:00 pm my_backup .backslash..backslash.server.backslash.share and not at 1:00 pm my_backup x: where x: is a connection made by the user.

Detailed Description Paragraph Table (83):
(Severity int, AckedRed int, AckedGreen int, AckedBlue int, UnackedRed int, UnackedGreen int, UnackedBlue int, unique(Severity), permanent); - The following database and table is required for the additional features - to support the Internet Service Monitors

Detailed Description Paragraph Table (90):
use database auto; insert into triggers values ('FindSpecifcSerial','select * from alerts.status whereSerial >= 4691 andSerial <= 4693;',",",",'DebugChangeField',",10431,0,1,60,0,0,1,",",",");insert into triggers values ('DetectUnknownServices','select * from service.status where LastReportAt < getdate - 3600;',",",",'SetServiceUnknown',",0,0,0,600,0,0,1,'This trigger detects services which have not been reported forthe given period. The associated action sets these services intothe unknown state. It is only needed with the Internet Service Monitors.',",",",); insert into triggers values ('CleanDetailsTable','delete from alerts.details where Identifier not in

```
(·(select Identifier from alerts.status));',",",",",",0,1,1,601,0,0,0,'This is a standard
automation for clearing old entries from the details table.',",","," ); insert into triggers
values ( 'DeleteAllEvents','delete from alerts.status where Serial <
4700;',",",",",",",10431,0,1,60,0,0,1,",",","," ); insert into triggers values
( 'AGetEventsToNotify','select * from alerts.status where NotifyPending =
1;',",",",",'TakeNotifyAction',",10431,1,1,1,0,0,1,",",","," ); insert into triggers values
( 'SelectLoggedToDbse','select * from alerts.status where
LoggedToDatabase=0andDatabaseElementKey<>0; ',",",",",'ChangeLoggedToDbse',",
0,0,1,5,0,0,1,",",","," ); insert into triggers values ( 'EscalateOff','update alerts.status set
Flash = 0, Grade = 0 where ((Flash = 1 or Grade > 0) and Acknowledged = 1) or (Severity =
0);',",",",",",",0,0,1,6,0,0,0,'Will set Flash field to 0 (not flashing) and Grade to 0 (not
escalated in this example) when an event that has previously had the Flash field set to 1 or
greater is either Acknowledged or Cleared (Severity = 0). ',",","," ); insert into triggers
values ( 'TimeKeyLogged','select * from alerts.status
whereTimeKeyLogged=0;',",",",",'GenerateTimeKey',",0,1,1,5,0,0,1,'99Overview: This trigger
selects all records from the alert_status table who have a TimeKeyLogged value equal to zero.
It then takes these records and runs the script TimeKeyGen','erator for each row.',"," );
insert into triggers values ( 'FindAlertStops','select * from alerts.status where AlertType =
2;',",",",",'RemoveAlertStops',",0,1,1,1,0,0,1,'Overview:',",","," ); insert into triggers values
( 'FlashNotAck','update alerts.status set Flash = 1, Grade = 1 where Flash = 0 and Acknowledged
= 0 and Severity = 5 and FirstOccurrence <= (getdate - 600);',",",",",",",0,0,1,31,0,0,0,'Will
set Flashing on (Flash=1) for events that are Critical (Severity=5)and are 10 minutes old but
haven.backslash.'t been acknowledged by a user yet(Acknowledge = 0). It sets Grade to 1 as a
further indication of the events escalation status.',",","," ); insert into triggers values
( 'GenericClear','select * from alerts.status where Type = 2 and Severity >
0;',",",",",'GenericClear',",10431,0,1,5,0,0,1,'This is a standard Automation for correlating two
problem/resolutionevents, ie correlating a Device Up event with a Device Down event. This is
done by checking the contents of the following fields;Type 1=Problem event, 2=Resolution
eventLastO','ccurrence ensure resolution is more recent then problemAlertGroup same Type of
event, ie Device Up/DownManager same source, same Probe reported both eventsNode same device
reported both eventsAlertKey same sub-device (link, disk pa','rtition etc) reported both
events'," ); insert into triggers values ( 'MailOnCritical','select * from alerts.status where
Severity = 5 and Grade < 2 and Acknowledged = 0 and LastOccurrence <= (getdate - (60 *
30));',",",",",'MailOnCritical',",0,0,1,33,0,0,1,'Finds all events which are Critical
(Severity=5) that are 30 minutes oldbut still haven.backslash.'t been Acknowledged (or
escalated to level 2, Grade=2).
The .backslash.'via.backslash.'@Identifier.backslash.'.backslash.' command is used The Action
sets Grade = 2 to show the new escalation status',".
The .backslash.'via .backslash.'@Identifier.backslash.'.backslash.' command is used for
improved lookup performance. The Action then activates the external script
$OMNIHOME/utils/nco_mail and passes data from the event to that script. The script is a simple
script which will insert t','he events data into a mail message and mail to a user (in this
example .backslash.'root.backslash.' user on the local machine). NOTE: This tool is UNIX
specific unless an equivalent NT mailer is available.'," ); insert into triggers values
( 'FindEventsToRegen1','select * from alerts.status whereAlertType = 1 andDuration = 0
andFirstOccurrence > getdate - (hourofday*60*60) - (minuteofhour*60) - 86400 - 60
andFirstOccurrence < getdate - (hourofday*60*60) - (minuteofhour*60) -
60;',",",",",'ModifyAlertToRegen',",10431,1,1,1,0,0,1,",",","," ); insert into triggers values
( 'FindEventsToRegen2','select * from alerts.status whereSustainedAlert =
2;',",",",",'InsertRegenAlert',",0,1,1,1,0,0,1,",",","," ); insert into triggers values
( 'CalculateDuration','select * from alerts.status where AlertType = 1 andSeverity = 0
andDuration = - 1;',",",",",'SetDuration',",0,1,1,1,0,0,1,'Overview:',",","," ); insert into
triggers values ( 'Expire','select * from alerts.status where Type > 10 and Severity >
0;',",",",",'Expire',",10431,0,1,65,0,0,1,'This is a standard Automation for finding events that
have passed their.backslash.'Expire.backslash.' time (stored in the Type field). The Action
sets the events toClear (Severity 0).',",","," ); insert into triggers values
( 'CleanJournalTable','delete from alerts.journal where Serial not in ((select Serial from
alerts.status));',",",",",",",0,1,1,602,0,0,0,'This is a standard automation for clearing old
entries from the journal table.',",","," ); insert into triggers values ( 'DeleteClears','delete
from alerts.status where Severity = 0 and StateChange < (getdate -
129600);',",",",",",",10431,1,1,67,0,0,0,'This is a standard Automation for deleting Cleared
events from the ObjectServer. When using in conjunction with Internet ServiceMonitors (ISMs)
amend the where statement as follows;...where Severity = 0 and StateChange < (getdate - 120)
```

```
and Manage','rnot like.backslash.'ISM.backslash.';bloomje:I changed this to remove cleared
events after 36 hours of no state change, not two minutes',"," ); set recovery_sequence for
triggers to 0; -- DO NOT CHANGE OR REMOVE THIS LINE !
  .
```

Detailed Description Paragraph Table (91):

```
use database auto; insert into actions values ( 'InsertRegenAlert',1,'update
alerts.statussetSustainedAlert = 1, Severity = OwhereSerial = @Serial; insert into
alerts.status values(.backslash.'@RegenIdentifier.backslash.' ,
0 ,.backslash.'@Node.backslash.' ,.backslash.'@NodeAlias.backslash.' ,.backslash.'@Manager.backs
@Poll ,@Type ,0 ,@Class ,@Grade ,.backslash.'@Location.backslash.' ,@OwnerUID ,@OwnerGID ,@Ackno
1 ,
.backslash.'@NotifyAction.backslash.' ,@NotifyPending ,0 ,@DatabaseElementKey ,0 ,@OpenTicket ,.
insert into actions values ( 'GenerateTimeKey', 1,'update alerts.status setTimeKeyLogged=1
whereSerial=@Serial; ',",",",1 ,'/opt/ECM/bin/bloomje/GenerateTimeKey.pl',
'@LastOccurrence','twmmdb02',0,0,",",",", This script takes the field LastOccurrence and
translates it into day, month, year, hour, min, seconds.',",","," ); insert into actions values
( 'DebugChangeField',0,'update alerts.statussetSummary = .backslash.'Blooms
@Summary.backslash.'whereSerial = @Serial;',",", ",0,",",",0,0,",",",",",",","," ); insert into
actions values ( 'ModifyAlertToRegen', 1,'update alerts.statussetDuration = 86400 -
(@FirstOccurrence - @MidnightTime), AlertStopTime = @MidnightTime + 86400, RegenIdentifier
= .backslash.'@Node @EventCode @MidnightTime.backslash.', RegenMidnightTime = @MidnightTime +
86400, SustainedAlert = 2,',' DebugFieldChar = .backslash.'ModifyAlertToRegen.backslash.'
whereSerial = @Serial;',",",0,",",",0,0,",",",",",",",", ); insert into actions values
( 'SetDuration',1,'update alerts.status set LoggedToDatabase = 0,DebugFieldChar
= .backslash.'SetDuration just ran.backslash.',Identifier
= .backslash.'@Node.@EventCode.@FirstOccurrence.backslash.' , Duration = (@AlertStopTime -
@FirstOccurrence) whereSerial = @Serial;',",",",0,",",",0,0,",",",",",","," ); insert into
actions values ( 'GenericClear',0,'update alerts.status set Severity = 0 where Severity > 0 and
Type = 1 and LastOccurrence < @LastOccurrence and AlertGroup
= .backslash.'@AlertGroup.backslash.' and Manager = .backslash.'@Manager.backslash.' and Node
= .backslash.'@Node.backslash.' and AlertKey = .backslash.'@AlertKey.backslash.'; update
alerts.status set Severity = ','0 where Serial = @Serial; ',", ",0,",",",0,0,",",",",'This is a
standard Automation for correlating two problem/resolutionevents, ie correlating a Device Up
event with a Device Down event. This is done by checking the contents of the following fields;
Type 1=Problem event, 2=Resolution eventLastO', 'ccurrence ensure resolution is more recent
then problemAlertGroup same Type of event, ie Device Up/DownManager same source, same Probe
reported both eventsNode same device reported both eventsAlertKey same sub-device (link, disk
pa','rtition etc) reported both events'," ); insert into actions values
('MailOnCritical',1,'update alerts.status via .backslash.'@Identifier.backslash.'
setGrade=2;',",",",1,'$OMNIHOME/utils/ nco_mail','@Node @Severity NCO_MAIL_MESSAGE
root@omnihost .backslash.'@Summary.backslash.",'omnihost',0,0,",",",",",",",",); insert into
actions values ( 'RegenAlerts',0,'update alerts.statussetSeverity = 0,Duration = 86400 -
(@LastOccurrence - @MidnightTime),AlertStopTime = @MidnightTime + 86400,TempChar
= .backslash.'@OldIdentifier (@MidnightTime).backslash.',TempInt = @MidnightTime + 86400,
DebugFieldChar = .backslash.'Reg','enAlerts ran on me.backslash.'whereSerial =
@Serial;',",",0,",",",0,0,",",",",",","," ); insert into actions values
( 'SetServiceUnknown',1,'svc update .backslash.'@Name.backslash.'
3;',",",",0,",",",0,0,",",",",'This action sets the state of the given service to be Unknown.
It is only needed with the Internet Service Monitors.',",","); insert into actions values
( 'TakeNotifyAction', 1 ,'update alerts.statussetNotifyPending = OwhereSerial =
@Serial;',",",",1,'/opt/ECM/utils/notify_spooler.pl',
'.backslash."@Node.backslash." .backslash."@EventCode.backslash." .backslash."@LastOccurrence.ba
'twmmdb02',0,0,",",",",",","," ); insert into actions values ( 'ChangeLoggedToDbse',0,'update
alerts.status set LoggedToDatabase=1whereSerial=@Serial; ',",",",0,",",",0,0,",",",",",","," );
insert into actions values ('Expire',1,'update alerts.status set Severity = 0 where Serial =
@Serial and LastOccurrence < (getdate - @Type);',",",",0,",",",0,0,",",",",'This is a standard
Automation for finding events that have passed their.backslash.'Expire.backslash.' time (stored
in the Type field). The Action sets the events toClear (Severity 0).',",","," ); insert into
actions values ( 'RemoveAlertStops',1,'update alerts.status set Severity = 0, AlertStopTime =
@FirstOccurrence, DebugFieldChar = .backslash.'RemoveAlertStops ran.backslash.' where Severity
<> 0 and AlertType = 1 and Node = .backslash.'@Node.backslash.' and EventCode = @EventCode - 1
','and LastOccurrence <= @LastOccurrence; delete from alerts.status where Serial = @Serial;
',",",0,",",",0,0,",",",",",","," ); set recovery_sequence for actions to 0; -- DO NOT CHANGE
```

OR REMOVE THIS LINE!

Detailed Description Paragraph Table (112):

TABLE 39 Service Assurance Application Memory and System Requirements Suggested Memory Program Requirements Suggested System Requirements Collector RAM: Hardware Requirements: Object Server: 50 SUN Workstations: will run all components of Collector. Mbytes per server SPARC 20 or better, Ultra 1 or better, with appropriate (dependant upon RAM and disk. (recommended system for desktop only: number of events SPARC 5 or better. in the Object HP Workstations: will run all components of Collector. server) C110 workstation or better, D230, K210, T520 servers or Desktops: better, with appropriate RAM and disk. 10 Mbytes per AIX Workstations: AIX .upsilon.3.2 will run Generic, NetView desktop (standard and Syslog probes, with appropriate RAM and disk. AIX user), 15 Mbytes 4.1.2 and above will run all 3.2.1 components. per desktop NT Workstations: NT probes and EventList at present, (standard user plus with appropriate RAM and disk. objective view), 20 Disk Space: Mbytes per The following table lists the amount of disk storage required to desktop (above store components of the Collector system. In some cases, plus administrator further economies may be possible, i.e., by discarding unused tools) probes. Probes: 5 Mbytes JEL per probe Common Object Process Gate Probes daem Gateways: 15 Platform/OS Files Desktop Server Control way (all) on Mbytes per SunOS 4.1.x 0.7 M 30 M 1.8 M 1 M 1.5 M 19 M 7 M gateway Sun Solaris 0.9 M 12 M 2.5 M 1.4 M 2.0 M 23 M 7 M Java Event List: 5 2.x Mbytes per HP-UX 9.07 0.8 M 11 M 1.9 M 1 M 1.6 M 18 M 7 M daemon HP-UX 0.8 M 11 M 2.0 M 1 M 1.6 M 16 M 7 M Web server: 2.5 10.10 and Mbytes per user 10.20 typical web AIX 0.8 M -- -- 1.5 M -- 4 M -- server, will vary) Windows -- -- -- -- -- 6 M -- NT In addition 10-20 Mb should be allowed for logging space on systems running Gateways, Object Servers, Probes or Process Control. Reporter It is recommended that you run the Collector server with a minimum of 128 MB and a maximum of 256 MB of memory. These values recommendations but if you have the potential for large amounts of data to report on, this is going to be vital to the efficiency of the application. This section describes what action to take on the server if you do not have these values set and you are getting memory errors while running Collector/ Reporter. See page 32 of the Admin/user guide of Reporter for making adjustments to memory for UNIX. PATROLL The CPU and Operating System Requirements: ERLER PATROLLERLER Sun SPARC; Solaris2; Min version 2.4; Solaris2-sun4 Console should be run Sun SPARC; Solaris2; Min version 2.5; Solaris25-sun4 on a machine with at nls is a system prerequisite for Sun O/S 4.xx installations least 64 MB of Disk Space: memory. Each PATROLLERLER Console requires about 20 MB of disk space. The Console also requires an additional 31 MB of disk space for the supporting files such as icon images and online help files. You will need an additional 27 MB of disk space if you choose to install the optional background images for European country maps. Each PATROLLERLER Agent requires about 10 MB of disk space. Each PATROLLERLER Event Manager (PEM) Console requires about 5 MB of disk space. If the PEM Console is installed independently of the PATROLLER Console, then an additional 24 MB of disk space is required for the supporting files such as icon images and online help files. PATROLLERLER Module space requirements vary. The installation script furnishes an estimate of each module's requirements: Event ii. Hardware Configuration: (minimum UNIX) Correlator 48 MB for the 200 MB disc space color monitor 1024 .times. 768 and server Solaris 2.5.1 or HP/UX 10.20 Manager 32 MB for the Hardware Configuration: (minimum NT) client P5-166 Intel Processor, 40 MB disc space, color monitor NT 1024 .times. 768 32 MB RAM Microsoft Windows NT 4.0 Note: Supports the following: OpenView Network Node Manager - versions 4.11 and 5.01 OpenView IT/Operations - version 4.0 for HP-UX HP iii. Unix OpenView 64 Mbytes Computer: Unix recommended Use one of the following computers as the NNM Version minimum Management Station. 5.01 32 Mbytes HP 9000 Series 700 NT minimum for HP 9000 Series 800, J and K models Version NNM 250 Sun SPARCstation 5,10,20,2000 5.02 Note: The amount of Sun SPARCclassics RAM in your NNM Sun Servers management station Graphics Dipslay should be based on the X Terminal or Workstation graphics display with number of nodes 1280 .times. 1024 resolution, 8 color planes (recommended) which you wish to 1024 .times. 768 resolution, 6 color planes (minimum) manage. Additional 20" display RAM may also be Installation Device required to run third- CD-ROM drive party OpenView Disk Space applications on top of The minimal disk space for NNM installation is shown NNM. See the below Network Node HP-UX 9.x - 85 Mbytes Manager Performance HP-UX 10.x - 85 Mbytes and Configuration Solaris 2.x - 130 Mbytes Guide for assistance in Operating System calculating for the One of the operating systems listed below must be running optimum amount of on the NNM mgmt system. RAM. HP-UX 9.0-9.07 (9.x) iv. HP-UX 10.01, 10.10, and 10.20 (10.x) 32 Mbytes of Solaris 2.4 and 2.5.x RAM to manage Networking Subsystem 250 nodes and, The appropriate TCP/IP networking subsystems (e.g. LAN 48 Mbytes of Link, ARPA Services) found within the operating system RAM to manage must be installed and configured to yield TCP/IP network up to 2500 nodes. connectivity Note: You will need connectivity. to have a minimum Windowing Subsystem amount of paging file HP-UX: X Windows/Motif size (available virtual

Solaris: OpenWindows memory) configured. SNMP Agent If NNM is being The NNM management station must be running an SNMP installed as a remote agent. An SNMP agent is shipped with NNM for HP-UX 9.x console, Paging Files and Solaris systems, and is automatically installed when is checked to be at installing NNM. HP-UX 10.x systems use the SNMP agent least 50 Mbytes. If shipped with the operating system. this is not a remote NT console installation, Operating System 60 Mbytes will be the You should be running Windows NT 3.51 or Windows NT minimum. 4.0 for NNM or higher to run successfully Graphics Display Your screen resolution must be at least 600 .times. 800 to support NNM display objects. Networking Subsystem You should have TCP/IP services installed. Platinum v. VCI (important component of CCC/Harvest) Technology Server Microsoft Windows 95 or Windows NT CCC/Harvest 16 Mb main Tool that supports Microsoft's Common Source Code memory Control (SCC) Interface. Following is a partial list of It is recommended SCC-compliant tools: that 2 Mb of Visual C++ 4.2 and 5.0 virtual memory is Visual Basic 4.0 and 5.0 allocated for each Visual J++ 1.1 user Paradigm Plus 3.5.1 Client (Solaris) PowerBuilder 5.0.03 SPARCstation or Unix SPARCserver CD-ROM drive, 8 mm tape drive, 4 mm DDS cartridge, or running Solaris 2.5 1/4 inch cartridge tape drive (SunOS 5.5) or Oracle RDBMS version 7.3 or beyond, including the with X-Windows following options: System Version SQL*Plus, PL/SQL, SQL*Loader, Pro-C, SQL*Net 11R5. Note: HP-UX 10 requires Oracle 7.3.4 or beyond. Approximately 50 NT MB of disk space IBM-compatible computer with a 486, or Pentium is required for the processor installation process Network connection to a Unix or Windows NT-based of the server using the TCP/IP protocol CCC/Harvest product files. vi. Server At least 12 Mb of free hard drive space A minimum of 32 Mb main memory. The Oracle database and CC/Harvest Broker together require about 14

CLAIMS:

1. A method for providing service assurance for a network to maintain a predetermined agreed upon Quality of Service, comprising the steps of: (a) generating an alarm to indicate a status of a network; wherein the step of generating an alarm to indicate a status of a network further comprises the steps of: selecting a parameter of the network that is to be monitored, determining a triggering level of the parameter, monitoring the parameter of an occurrence of the triggering level, and initiating an alarm notification upon the monitored occurrence of the triggering level; (b) dispatching network event information of the network upon generation of the alarm; (c) mapping the network event information; (d) manipulating data collected on the status of the network, wherein manipulating data comprises: (i) concatenating data collected on a network into a master file; (ii) reformatting the concatenated data into a standardized format; (iii) translating the standardized data to key codes; (iv) sorting the translated data according to predetermined criteria; and (v) concatenating the sorted data together; (e) storing the manipulated data in a database; and (f) graphically conveying availability of the network.

2. A method as recited in claim 1, wherein the step of dispatching network event information of the network upon generation of the alarm further comprises the steps of: monitoring a network for an event; generating at least one notification action based upon the occurrence of the event, wherein the notification action comprises at least one of: an alphanumeric page, an e-mail message, a resolution script, a remedy trouble ticket, and a log message; and transmitting the notification action to notify a recipient about the occurrence of the event.

3. A method as recited in claim 1, wherein the step of mapping the network event information further comprises the steps of: monitoring a network for the occurrence of availability events, threshold events, and trap events, correlating at least one occurred event to at least one other occurred event to generate at least one correlating event, mapping the occurred events and correlating events on at least one network map; and displaying the network map.

4. A method as recited in claim 1, wherein the step of graphically conveying availability of the network further comprises the steps of: selecting report parameters relating to availability of monitored elements, services, and processes of a network, polling a database for data that matches the report parameters, generating a graph from the data that matches the report parameters, and displaying the generated graph to graphically represent the monitored elements, services, and processes of the network.

5. A computer program embodied on a computer readable medium for providing service assurance for a network to maintain a predetermined agreed upon Quality of Service, comprising: (a) a code segment for generating an alarm to indicate a status of a network; wherein the code

segment for generating an alarm to indicate a status of a network is further adapted for selecting a parameter of the network that is to be monitored, determining a triggering level of the parameter, monitoring the parameter of an occurrence of the triggering level, and initiating an alarm notification upon the monitored occurrence of the triggering level; (b) a code segment for dispatching network event information of the network upon generation of the alarm; (c) a code segment for mapping the network event information; (d) a code segment for manipulating data collected on the status of the network, wherein manipulating data comprises: (i) concatenating data collected on a network into a master file; (ii) reformatting the concatenated data into a standardized format; (iii) translating the standardized data to key codes; (iv) sorting the translated data according to predetermined criteria; and (v) concatenating the sorted data together; (e) a code segment for storing the manipulated data in a database; and (f) a code segment for graphically conveying availability of the network.

6. A computer program as recited in claim 5, wherein the code segment for dispatching network event information of the network upon generation of the alarm is further adapted for monitoring a network for an event; generating at least one notification action based upon the occurrence of the event, wherein the notification action comprises at least one of: an alphanumeric page, an e-mail message, a resolution script, a remedy trouble ticket, and a log message; and transmitting the notification action to notify a recipient about the occurrence of the event.

7. A computer program as recited in claim 5, wherein the code segment for mapping the network event information is further adapted for monitoring a network for the occurrence of availability events, threshold events, and trap events, correlating at least one occurred event to at least one other occurred event to generate at least one correlating event, mapping the occurred events and correlating events on at least one network map; and displaying the network map.

8. A computer program as recited in claim 5, wherein the code segment for graphically conveying availability of the network is further adapted for selecting report parameters relating to availability of monitored elements, services, and processes of a network, polling a database for data that matches the report parameters, generating a graph from the data that matches the report parameters, and displaying the generated graph to graphically represent the monitored elements, services, and processes of the network.

9. A system for providing service assurance for a network to maintain a predetermined agreed upon Quality of Service, comprising: (a) logic for generating an alarm to indicate a status of a network; wherein the logic for generating an alarm to indicate a status of a network is further adapted for selecting a parameter of the network that is to be monitored, determining a triggering level of the parameter, monitoring the parameter of an occurrence of the triggering level, and initiating an alarm notification upon the monitored occurrence of the triggering level; (b) logic for dispatching network event information of the network upon generation of the alarm; (c) logic for mapping the network event information; (d) logic for manipulating data collected on the status of the network, wherein manipulating data comprises: (i) concatenating data collected on a network into a master file; (ii) reformatting the concatenated data into a standardized format; (iii) translating the standardized data to key codes; (iv) sorting the translated data according to predetermined criteria; and (v) concatenating the sorted data together; (e) logic for storing the manipulated data in a database; and (f) logic for graphically conveying availability of the network.

10. A system as recited in claim 9, wherein the logic for dispatching network event information of the network upon generation of the alarm is further adapted for monitoring a network for an event; generating at least one notification action based upon the occurrence of the event, wherein the notification action comprises at least one of: an alphanumeric page, an e-mail message, a resolution script, a remedy trouble ticket, and a log message; and transmitting the notification action to notify a recipient about the occurrence of the event.

11. A system as recited in claim 9, wherein the logic for mapping the network event information is further adapted for monitoring a network for the occurrence of availability events, threshold events, and trap events, correlating at least one occurred event to at least one other occurred event to generate at least one correlating event, mapping the occurred events and correlating events on at least one network map; and displaying the network map.

12. A system as recited in claim 9, wherein the logic for graphically conveying availability of

the network is further adapted for selecting report parameters relating to availability of monitored elements, services, and processes of a network, polling a database for data that matches the report parameters, generating a graph from the data that matches the report parameters, and displaying the generated graph to graphically represent the monitored elements, services, and processes of the network.

Previous Doc Next Doc Go to Doc#